66. Plus One

Easy

Given a **non-empty** array of digits representing a non-negative integer, plus one to the integer.

The digits are stored such that the most significant digit is at the head of the list, and each element in the array contain a single digit.

You may assume the integer does not contain any leading zero, except the number 0 itself.

**Example 1:**

**Input:** [1,2,3]  
**Output:** [1,2,4]  
**Explanation:** The array represents the integer 123.

**Example 2:**

**Input:** [4,3,2,1]  
**Output:** [4,3,2,2]  
**Explanation:** The array represents the integer 4321.  
C

/\*\*

\* Return an array of size \*returnSize.

\* Note: The returned array must be malloced, assume caller calls free().

\*/

int\* plusOne(int\* digits, int digitsSize, int\* returnSize) {

int\* ret = (int\*)malloc(sizeof(int)\*(digitsSize+1));

bool carry=true;

for(int i=digitsSize;i>0;i--){

ret[i]=(carry==true)?digits[i-1]+1:digits[i-1];

if(ret[i]>9){

ret[i]=ret[i]%10;

}else carry=false;

}

if(carry) {

ret[0]=1;

\*returnSize=digitsSize+1;

return ret;

}

\*returnSize=digitsSize;

return ret+1;

}

Success

[Details](https://leetcode.com/submissions/detail/206603224/)

Runtime: 4 ms, faster than 8.47% of C online submissions for Plus One.

Memory Usage: 4.1 MB, less than 0.00% of C online submissions for Plus One.

C++

class Solution {

public:

vector<int> plusOne(vector<int>& digits) {

bool carry=true;

for(int i=digits.size()-1;i>=0&&carry;i--){

digits[i]++;

if(digits[i]>9){

digits[i]=digits[i]%10;

}else carry=false;

}

if(carry) digits.insert(digits.begin(),1);

return digits;

}

};

Success

[Details](https://leetcode.com/submissions/detail/206599335/)

Runtime: 4 ms, faster than 100.00% of C++ online submissions for Plus One.

Memory Usage: 4.6 MB, less than 0.49% of C++ online submissions for Plus One.